Logs and Log Merges

In an Adabas cluster environment, the protection logs (and optionally, the command logs) of all individual nuclei in the cluster are merged into single log files in chronological order for the cluster database shared by all the nuclei as a whole. The chronological order is determined by timestamps on all individual private protection and command records. Once the private records are merged, the timestamps are removed. The timestamp is the only difference between an unmerged (that is, private) and merged record.

Given that each cluster nucleus has its own PLOG data sets, checkpoints are no longer identified only by their name, PLOG number, and PLOG block number, but also by the NUCID of the nucleus that writes the checkpoint.

ADARES is the only utility that handles both the unmerged and merged PLOGs and CLOGs. The log merge process:

- merges PLOG or CLOG records into correct chronological sequence.

- avoids a single point of failure by partially merging as much PLOG or CLOG data as possible if a nucleus is unable to respond to a switch.

- is automatic; merges directly from the dual or multiple PLOGs or CLOGs, which eliminates the need to retain control information for tracking unmerged logs.

- quickly copies and merges so that the PLOG or CLOG can be reused.

- uses an identical PLOG or CLOG record layout for an unmerged or a merged record.

- produces a consistent amount of data from each PLCOPY or CLCOPY.

For more detailed information, read *Adabas Parallel Services Operations* and read about the ADARES utility in the Adabas utilities documentation.

This chapter covers the following topics:

- Protection Logs

- Command Logs

- Coordinated Log Switching

Protection Logs

Either all nuclei in an Adabas Parallel Services cluster must run without protection logs (PLOGs) or all nuclei must run with PLOGs. The same ADARUN PLOGRQ setting is enforced for all nuclei in a cluster. This value cannot be modified during a session.

If the nuclei in the cluster are running with PLOGs, they must be dual or multiple PLOGs: sequential PLOGs are not allowed. For more information, read your Adabas operations documentation.

Each nucleus in the cluster writes its protection data to its own Work and PLOG data sets: no synchronization is required. However, every protection record is preceded by an 8-byte timestamp indicating the time of its creation. The timestamp is the only difference between protection records written by a cluster nucleus and those written by a noncluster nucleus. The timestamp is required for the merge process and is removed after the protection log is merged but before it is written to the sequential data set.

This section covers the following topics:

- Merging Protection Logs

- Intermediate Data Sets

- User Exits for Merging Protection Logs

Merging Protection Logs

For recovery processing, all protection log data sets are automatically merged into a single log stream for each cluster database when an ADARES PLCOPY is executed in an Adabas Parallel Services nucleus cluster environment. The PLCOPY process accesses the parallel participant table (read *Parallel Participant Table (PPT)*) to determine which protection logs to copy and opens the appropriate data sets using dynamic allocation (read *Dynamic Allocation*). PLCOPY copies and merges as much data as possible; if a nucleus is still writing to a protection log data set, PLCOPY partially merges the data set.

ADARES expects that at least one of the protection logs being merged is at 'completed' status. If this is not the case, ADARES reports that there is no data to be copied.

The merge begins with the lowest timestamp from all protection logs being merged and ends with the lowest of the ending timestamps from all data sets. Records beyond this point are written to an 'intermediate' data set, which must be supplied as input to the subsequent merge.

Intermediate Data Sets

It is necessary to supply two intermediate data sets to the PLCOPY merge process: one is used as input and the other as output. ADARES analyzes both data sets to determine which is which. A cross-check ensures that the correct input intermediate data set has been supplied; the utility will not continue if the correct data set has not been supplied. The input intermediate data set contains "leftover unmerged" data from the last run of the utility, without which the merged protection data would be incomplete.

User Exits for Merging Protection Logs

A user exit is necessary for merging protection logs automatically: user exit 2 (UEX2) for dual logs or user exit 12 (UEX12) for multiple logs. However, you may choose to run without the user exit. This may be advisable for testing purposes, for example. If one nucleus in a cluster has the user exit specified, then all other nuclei in the cluster must also supply the user exit. The user exit supplied must be the same across all nuclei.

Sample user exits (USEREX2P and UEX12) are provided that illustrate the necessary changes for the intermediate data sets/files. Once the Link (DD) statements for the PLOG files/data sets have been supplied on the session startup JCL, you do not need to supply them again for ADARES as these are opened using dynamic allocation. If the Link (DD) statements are supplied, they are ignored.

If there is a problem with the user exit, the Extended Error Recovery logic allows you to disable the exit and load a new exit without bringing the nucleus down.

Command Logs

In order to log commands during a session, a nucleus requires a command log data set. Command logging is activated by setting the ADARUN parameter LOGGING=YES.

Dual or multiple command logging requires that you set ADARUN parameters correctly. For more information about the ADARUN utility, read your Adabas operations documentation.

This section covers the following topics:

- Merging Command Logs Manually

- Merging Command Logs Automatically

- Dynamically Modifying the CLOGMRG Parameter Setting

Merging Command Logs Manually

You can copy the dual or multiple command logs into a single sequential data set for a nucleus using the default ADARES CLCOPY function. Then, using the output from the CLCOPY execution as input, you can use the ADARES MERGE CLOG function to create a single cluster-wide command log containing, in chronological order, all Adabas commands executed by any of the cluster nuclei in the time period covered by the log:

```
ADARES MERGE CLOG NUMLOG=n
```

where n is the number of command logs to be merged, but not more than 32.

The ADARES MERGE CLOG function takes as input the sequential command log data sets/files from each of the cluster nuclei that were produced by running CLCOPY, merges them and copies the resulting command log data set from disk to a sequential data set.

Merging Command Logs Automatically

For accounting or other tracking purposes, you may want to automate the CLOG merge process. If you are using dual or multiple command logging, you set the ADARUN parameter CLOGMRG=YES, and if the correct user exit is available (user exit 2 for dual logs or user exit 12 for multiple logs), CLOGs are merged automatically as part of the CLCOPY process in much the same way that PLOGs are merged as part of the PLCOPY process. The ADARUN parameter LOGGING=YES must also be specified.

Existing CLCOPY jobs must be modified to include intermediate data sets/files. The sample CLCOPY job ADARESMC illustrates the necessary addition of the intermediate data sets/files. When intermediate data sets/files are used for both PLCOPY and CLCOPY merge processes, the data set names must be unique so that they are not overwritten.

You can use the same exit for the CLOG merge process that is supplied for the PLOG merge process (see the sample exit USEREX2P or UEX12):

- When CLOGMRG=YES, use a single user exit 2 or user exit 12 for both PLOG and CLOG merging; both PLOG and CLOG data sets/files are dynamically allocated. Data sets/files specified in the user exit JCL are ignored.

- When CLOGMRG=NO (the default), CLOG data sets/files are *not* dynamically allocated and must be specified in the ADARES JCL; therefore, user exit 2 or user exit 12 must submit different CLCOPY jobs that are appropriate for the nucleus that calls the exit.

  Note:
  If CLOGMRG=NO, you can manually merge CLOGs across a cluster.

Likewise, CLOG data sets/files with slightly modified names can be included in the user exit 2 or user exit 12 JCL used for other NUCIDs.

Dynamically Modifying the CLOGMRG Parameter Setting

The CLOGMRG parameter setting can be changed dynamically with the new operator command CLOGMRG, the new ADADBS OPERCOM CLOGMRG function, or with the Modify Parameters function of Adabas Online System.

CLOGMRG is a global (that is, cluster-wide) parameter. A change made to one nucleus is automatically propagated to all nuclei in the cluster.

Coordinated Log Switching

The log merge process (PLOG or CLOG) makes it necessary to coordinate the log switching process so that the merge process can copy as much data as possible without causing too many switches.

The first cluster nucleus that needs to switch logs (when a log is full or the FEOFPL command for PLOGs or the FEOFCL command for CLOGs has been issued) sends a broadcast message to the other nuclei in the cluster indicating that it is switching. It then goes on as normal, switching and calling the appropriate user exit.

Note:
User exit 2 for dual logs or user exit 12 for multiple logs is recommended and is required if PLOGRQ=FORCE is set. If it is supplied to one nucleus in a cluster, it must be supplied to all the other nuclei. PLOGRQ must have the same setting across all nuclei in the cluster. If running with PLOGs, only dual or multiple PLOGs are allowed.

When a cluster nucleus receives the message that another nucleus is switching logs, it must decide if it will switch based on the current status of its current log and whether it has an available log data set:

- If it switches, it does so without calling user exit 2 (dual logs) or user exit 12 (multiple logs).

- If it does not switch, it forces a log write of the current block so that ADARES has as much data as possible.

ADARES briefly waits at the start to give all nuclei time to switch. If a nucleus does not switch logs, ADARES copies as much data as possible from the active log (partial merge).

This section covers the following topics:

- Global FEOFPL and FEOFCL Commands

Global FEOFPL and FEOFCL Commands

The GLOBAL option has been added to both the ADADBS OPERCOM FEOFPL and FEOFCL commands:

- When the GLOBAL option is not specified, consideration is given to how full the PLOG or CLOG is before an attempt is made to switch them.

- When the GLOBAL option is specified, a log switch is forced for each nucleus in the cluster if there is an available log data set to write to. This may be useful in situations where all nuclei need to switch at the same time to capture all merged records in the output data set.

If a log data set to write to is not available, the switch does not occur even if GLOBAL has been specified. This functions the same way when the FEOFPL or FEOFCL command is issued for a single nucleus.